LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-641855

An accurate and scalable O(N) algorithm for First-Principles Molecular Dynamics computations on large parallel computers

D. Osei-Kuffuor, J. L. Fattebert

August 5, 2013

Physical Review Letters

Disclaimer

An accurate and scalable $\mathcal{O}(\mathcal{N})$ algorithm for First-Principles Molecular Dynamics computations on large parallel computers

Daniel Osei-Kuffuor and Jean-Luc Fattebert
Center for Applied Scientific Computing, L-561, Lawrence Livermore National Laboratory, Livermore, CA 94551
(Dated: August 2, 2013)

We present the first truly scalable First-Principles Molecular Dynamics algorithm with O(N) complexity and fully controllable accuracy, capable of simulating systems of sizes that were previously impossible with this degree of accuracy. By avoiding global communication, we have extended W. Kohn's condensed matter "nearsightedness" principle [1] to a practical computational scheme capable of extreme scalability. Accuracy is controlled by the mesh spacing of the finite difference discretization, the size of the localization regions in which the electronic wavefunctions are confined, and a cutoff beyond which the components of the overlap matrix can be omitted when computing selected elements of its inverse. We demonstrate the algorithm's excellent parallel scaling for up to 101,952 atoms on 23,328 processors, with a wall-clock time of the order of one minute per molecular dynamics time step and numerical error on the forces of less than $7 \cdot 10^{-4}$ Ha/Bohr.

Predictive materials simulation is becoming ever more important in a variety of fields. Density Functional Theory (DFT)[2, 3] simulations are widely used and are already helping in understanding, designing, controlling and manufacturing advanced materials [4]. First-Principles Molecular Dynamics (FPMD) is a general and fundamental predictive tool to study matter at atomistic scale. FPMD typically uses the Born-Oppenheimer approximation and requires solving the equations of DFT, the Kohn-Sham (KS) equations to obtain the electronic structure and calculate forces acting on the atoms. FPMD simulations, however, are limited to a few hundred atoms for tens of picoseconds due to their computational cost and the $O(N^3)$ complexity of typical DFT solvers. In addition, because each electronic wavefunction spreads over the whole simulation domain, and because values such as dot products between pairs of wavefunctions are needed to solve the KS equations, these calculations involve many global operations. This is a problem for large parallel computers where a key to fast simulation is to reduce global communications. Advanced $O(N^3)$ algorithms and implementations have been developed to distribute computational work efficiently on large parallel computers [5]. Pushing such a strategies on today largest computers has enabled very large calculations. Hasegawa et al.[6] have been able to simulate 107,292 atoms on one of the largest supercomputers in the world, but with a time to solution far too long to be used in FPMD applications.

Anticipating these issues, there has been a lot of research on the development of $O(N)$ complexity algorithms in the last two decades (for a recent review, see [7]). However, to make efficient use of large parallel computers, $O(N)$ complexity alone is not enough. Optimal algorithms need to also avoid global communications, a key hurdle which has not been addressed by most proposed $O(N)$ algorithms, making them difficult to scale beyond a few thousand processors. One category of algorithms with no major global communications is the so-called "Divide and Conquer" [8–10]. Dividing a problem into sub-problems can however be quite tricky. It can lead to hard-to-quantify errors [11].

To make effective use of unprecedented levels of concurrency and sheer number of processors on today's largest computers, new mathematical formulations and algorithms are needed. Algorithms with reduced complexity and better parallel scaling are essential to enable more realistic modeling on large high-performance computers. For example, having the capabilities of simulating tens of thousands of atoms on a routine basis will enable material scientists to model systems with interfaces, irregularities, large defects, or heterogeneous systems instead of perfect crystals. This will help bridge the gap between the atomistic scale and the meso and macroscopic levels of descriptions.

In this Letter, we present the first truly scalable FPMD algorithm with O(N) complexity and fully controllable accuracy, capable of simulating systems of sizes that were previously impossible. This extends W. Kohn's "nearsightedness" principle [1] in physical condensed matter systems to a computational "nearsightedness" principle. Our approach relies on the existence of well-localized generalized maximally localized Wannier functions (MLWF) [12], which exist in insulators with a finite band gap [13]. We use a real-space representation of the wavefunctions given by their values at each node of a uniform finite difference mesh[14], but other representations are possible. Exploiting the exponential decay of off-diagonal elements of the single particle density matrix is not a practical solution for density matrices obtained directly from these discretizations, because these matrices are too large and have too many non-zero elements. However, the dimension of these matrices can be reduced by first computing the solution of the KS equations on a mesh using orbitals confined to finite spherical regions[15, 16], and effectively obtain a minimal localized basis set similar to MLWF but with strictly localized orbitals. We then make use of decaying properties of matrices associated to this inter-

mediate basis set of localized functions.

Neglecting the spin, the electronic structure of a physical system made of $2N$ valence electrons is represented by N non-orthogonal wavefunctions $\{\phi\}_{i=1}^{N}$ which span the sub-space of occupied states. The electronic ground state minimizes the DFT energy functional

$$\begin{aligned} E_{KS}[\{\phi_i\}_{i=1}^{N}] &= \sum_{i,j=1}^{N} \left(S^{-1}\right)_{ij} \int_{\Omega} \phi_i(\mathbf{r}) \left(-\nabla^2\right) \phi_j(\mathbf{r}) d\mathbf{r} \\ &+ \frac{1}{2} \int_{\Omega} \int_{\Omega} \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 + E_{XC}[\rho] \\ &+ \sum_{i,j=1}^{N} 2 \left(S^{-1}\right)_{ij} \int_{\Omega} \phi_i(\mathbf{r}) (V_{ext}\phi_j)(\mathbf{r}) d\mathbf{r}. \end{aligned}$$

where the terms on the right side are the kinetic energy, the electrostatic energy, the exchange and correlation energy, and the potential energy of the electrons in the potential V_{ext} modeling the atomic cores. We use the pseudopotential approximation in its Kleinman-Bylander form [17] and the PBE exchange correlation functional[18]. The $N \times N$ matrix S is the overlap matrix, $S_{ij} = \int_{\Omega} \phi_i(\mathbf{r})\phi_j(\mathbf{r})$. There is no orthonormality constraint in this formulation. The wavefunctions have to remain non-degenerate and the matrix S^{-1} needs to be calculated and be consistent with the wavefunctions at every step of the iterative solution. In this formulation, the electronic density ρ appears in the form

$$\rho(\mathbf{r}) = 2 \sum_{i,j=1}^{N} (S^{-1})_{ij} \phi_i(\mathbf{r}) \phi_j(\mathbf{r}). \tag{2}$$

One can represent the solution of the minimization problem in Eq. 1 as a matrix Φ whose columns are made of the functions $\phi_i, i = 1, \ldots, N$. The advantage of this general non-orthogonal representation is that one can find transforms $\tilde{\Phi} = \Phi C$ such that $\tilde{\Phi}$ is sparse, or at least such that many of its coefficients can be neglected. Such a transform is given by the Maximally Localized Wannier functions [12], which minimize Eq.(1) but are much more localized than eigenfunctions. Instead of calculating this transformation — which requires $O(N^3)$ operations —, one can prescribe *a priori* strict confinement regions, one for each ϕ_i. Minimizing (1) under these constraints lead to a sparse solution very similar to MLWF, and which approximates well the exact solution for confinement regions large enough and centered at appropriate locations [19]. The number of degrees of freedom associated with each wavefunction is then $O(1)$ and the total number of degrees of freedom is $O(N)$ [15]. By confining each function to a strictly local region which does not grow with the problem size, all the operations to evaluate the energy functional (1) and its gradient become $O(N)$. The Hartree potential is solved by a multigrid preconditioned gradient algorithm, also a scalable $O(N)$ algorithm [20].

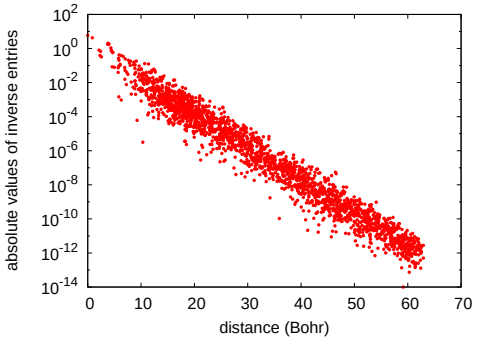

FIG. 1: Inverse entries in one column of S^{-1} as a function of the distance between this column and the orbitals associated with its nonzero entries (1888 atoms polymer sample).

Direct interactions between atoms through their atomic pseudopotentials are short-range and $O(N)$, using standard charge screening techniques [15]. Thus the only remaining $O(N^3)$ operation is the computation of S^{-1}.

We chose a domain decomposition approach, dividing the mesh evenly among all processors. In our implementation, each processor is responsible for the data associated to the local sub-domain, in particular the parts of the functions which overlap with the local mesh. It is also responsible for calculating partial dot product contributions from the local partial functions. While calculating S^{-1} requires $O(N^3)$ operations, it takes little time compared to other operations in a standard $O(N^3)$ approach, and its cubic scaling can even be ignored in $O(N)$ calculations for values of N up to a few thousands[14, 16]. However, one major issue is the global character of this computation: its exact calculation involves all the elements of S, which are distributed across all subdomains in a parallel framework. In other words, S^{-1} is a global coupling matrix, and its computation poses a major bottleneck for parallel scaling. From Eq.(1) and (2), it is clear that the only elements of S^{-1} needed are those of indices (i, j) corresponding to non-zero dot products between ϕ_i and ϕ_j, $-\nabla^2\phi_j$ or $V_{ext}\phi_j$. Other algorithms have been proposed in the literature for computing selected entries of the inverse of sparse symmetric matrices [21–23]. They rely on efficient Cholesky factorizations and divide-and-conquer strategies to compute exact entries of the inverse. However, they exhibit nonlinear complexity which will affect the overall complexity at large scale. The authors of [7] review a number methods for approximating the inverse of the Gram matrix for $O(N)$ electronic structure calculations. For most of these methods, linear scaling is achieved by enforcing some sparsity constraints on the matrix. However, the parallel implementations of these algorithms generally require some global coupling or frequent communication, which can affect parallel efficiency.

The matrix S is symmetric, positive definite and its eigenvalue spectrum is bounded independently of its size.

As a result, S^{-1} has off-diagonal coefficients that decay exponentially [24, 25]. Figure 1 gives an illustration of this result. Although S^{-1} is a dense matrix, this decay property allows for a sparse representation in which small terms can be dropped and numerical error can be systematically controlled with a single cutoff parameter. Our approach for computing S^{-1} is based on the approximate inverse strategy [26–28]. The general idea is to obtain a matrix M that satisfies

$$\arg \min_{M \in R^{N \times N}} \|SM - I\|$$

in the Frobenius norm, where I is the identity matrix. Typically, this is achieved by constructing from S, a smaller $k \times k$ matrix, $\hat{S}_j$, where $k \leq N$, and solving

$$\hat{S}_j \hat{m}_j = e_j$$

for each column j of M. Here, e_j represents the j-th column of I. The solution $\hat{m}_j$ expands into column j of M so that its non-zero entries approximates corresponding entries in S^{-1}. The matrix $\hat{S}_j$ is derived from a predetermined sparsity pattern imposed on the j-th column of M. For each column j of M, the non-zero pattern is obtained by considering only entries associated with local orbitals centered within some distance R_s, from the local orbital j. Thus R_s controls the block size k. A larger R_s yields a larger k and a more accurate approximation to S^{-1}. Moreover, since partial contributions to the matrix S are distributed across the processors, R_s also determines the extents from which to gather data to construct $\hat{S}_j$. An efficient communication strategy is used to gather partial contributions from adjacent processors. The algorithm processes data one direction at a time. Each processor communicates only with its two adjacent neighbors in one dimension, collecting and passing down data until all the data needed has been gathered. The process is then repeated in the other two dimensions. The result is that the cost of sending and receiving data from a 3D cluster of processors is reduced from the volume of the cluster, to the sum of the size of the cluster in each dimension. In our implementation, each processor only solves for the columns of M corresponding to the local orbitals centered on its local subdomain. Thus, one only needs to gather enough data to construct a single local block matrix $\hat{S}$ that can be used to solve for all the columns of M corresponding to these orbitals. An important observation is that for a given accuracy of the approximation to S^{-1}, the size of $\hat{S}$ is independent of the size of the global matrix S. Each linear system solve for the columns of M is done with a GMRES [29] accelerator, coupled with an ILU(0) preconditioner [30]. The preconditioner is constructed only once at the beginning of each MD step using the initial $\hat{S}$, and used throughout the iterative process to solve for the electronic structure in this MD step. GMRES convergence is typically achieved in five iterations or less.

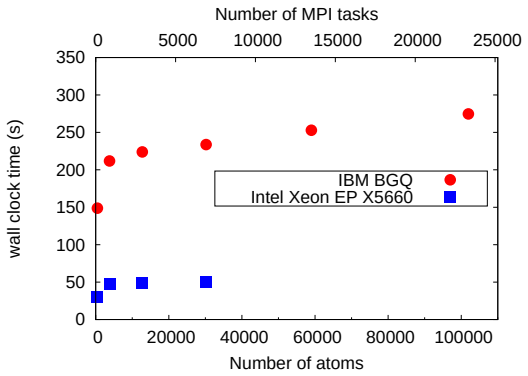

FIG. 2: Parallel weak scaling: wall clock time for 1 MD step on IBM BGQ architecture and Intel Xeon EP X5660 Linux cluster with High-speed interconnect (InfiniBand QDR QLogic) as a function of number of atoms and number of MPI tasks.

There is in principle no limit to parallel scaling for our algorithm. We demonstrate scalability up to 101,952 atoms on 23,328 cpus using the Lawrence Livermore National Laboratory IBM BGQ computer. We apply our algorithm to a polymer system with 1.05 g/cm^3 density. We start with a unit cell of size 30.568 Bohr containing 472 atoms, which results in 536 doubly occupied electronic orbitals. We replicate this system by a factor 2, 3, 4, 5 and 6 in each 3D direction to generate larger simulation cells and study parallel scaling. We scale up the number of atoms, while scaling up the number of processors at the same rate, that is, with a constant number of MPI tasks/atom (weak scaling study). Wall clock times for one MD step are shown in Fig. 2. One MD step typically involves about 15 updates for each wavefunction in our DFT iterative solution. For this particular test application, once a sufficiently large number of processors and atoms has been reached, each processor needs to communicate only within a neighborhood of 9×9×5 processors (sub-domains are not cubic). Using our communication algorithm of successive directions, that means a total number of 20 (8+8+4) sends and receives for each MPI task. This corresponds to gathering and inverting matrix blocks of sizes ≈ 2400.

There are three parameters to tune the accuracy of our calculations: the mesh spacing h, the confinement region radius R_c and the truncation radius for computing S^{-1}, R_S. We use a 4^{th} order finite difference scheme which leads to a discretization error $O(h^4)$ for the energy and forces[19]. The effect of confining orbitals in regions of radius R_c is assessed by comparing computed forces with the results of a full $O(N^3)$ calculation for the 472 atoms system, see Fig. 3. We observe an exponential decay of the error as a function of R_c. $R_c = 9$ Bohr — max. error of $6.4 \cdot 10^{-4}$ (Ha/Bohr) — is used to evaluate the additional error on atomic forces introduced by truncating elements when calculating S^{-1}. Fig. 4 displays that

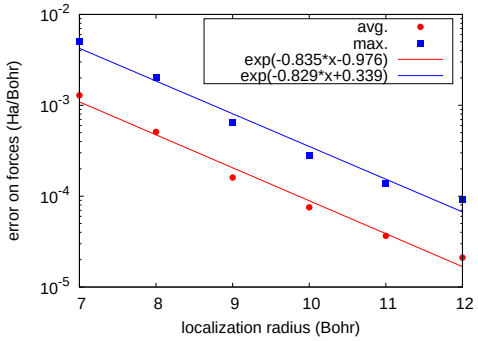

FIG. 3: Error on forces as a function of wavefunction confinement region radius R_c (average and maximum values over 472 atoms polymer sample).

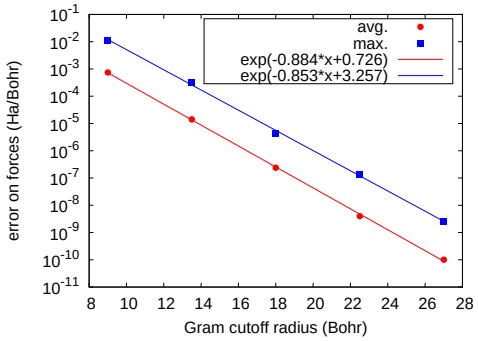

FIG. 4: Error on forces as a function of radius R_s, used to form each block in overlap inverse computation (average and maximum values over 12744 atoms polymer sample).

error as a function of the range within which matrix elements are gathered to approximate S^{-1}. The reference result we compare with is the one obtained with inverting the full matrix S, with $R_c = 9$ Bohr. We also observe an exponential decay of the error as a function of R_S. R_c= 9 Bohr and R_s= 18 Bohr lead to accurate forces and were used for the weak scaling study in Fig. 2.

At this time, we have a pure MPI implementation of our algorithm and the numbers in Fig. 2 correspond to a load close to the strong scaling limit. This already provides a time to solution close to one minute per MD step. Faster time to solution however are achievable by using threading, in particular on IBM BGQ architecture.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Work at LLNL was funded by the Laboratory Directed Research and Development Program under project tracking code 12-ERD-048. The authors would like to thank S. Hamel for sharing his data on polymers and for stimulating discussions, as well as E.W. Draeger for his careful reading of the manuscript and suggestions.

[1] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).
[2] P. Hohenberg and W. Kohn, Phys. Rev. B **136**, 864 (1964).
[3] W. Kohn and L. J. Sham, Phys. Rev. A **140**, 1133 (1965).
[4] A. Jain, G. Hautier, C. J. Moore, S. P. Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, Comput. Mat. Science **50**, 2295 (2011).
[5] F. Gygi, E. W. Draeger, M. Schulz, B. R. de Supinski, J. A. Gunnels, V. Austel, J. C. Sexton, F. Franchetti, S. Kral, C. W. Ueberhuber, et al., in *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing* (ACM, New York, NY, USA, 2006), p. 45.
[6] Y. Hasegawa, J.-I. Iwata, M. Tsuji, D. Takahashi, A. Oshiyama, K. Minami, T. Boku, F. Shoji, A. Uno, M. Kurokawa, et al., in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (ACM, New York, NY, USA, 2011), SC '11, pp. 1–11.
[7] D. R. Bowler and T. Miyazaki, Reports on Progress in Physics **75**, 036503 (2012).
[8] W. Yang, Phys. Rev. Lett. **66**, 1438 (1991).
[9] F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta, Phys. Rev. B **77** (2008).
[10] Z. Zhao, J. Meza, and L.-W. Wang, J.Phys.: Condens. Matter **20**, 294203 (2008).
[11] N. Ohba, S. Ogata, T. Kouno, T. Tamura, and R. Kobayashi, Comput. Phys. Commun. **183**, 1664 (2012).
[12] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997).
[13] C. Brouder, G. Panati, M. Calandra, C. Mourougane, and N. Marzari, Phys. Rev. Lett. **98** (2007).
[14] J.-L. Fattebert and F. Gygi, Phys. Rev. B **73**, 115124 (2006).
[15] G. Galli and M. Parrinello, Phys. Rev. Lett. **69**, 3547 (1992).
[16] J.-L. Fattebert and J. Bernholc, Phys. Rev. B **62**, 1713 (2000).
[17] L. Kleinman and D. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).
[18] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[19] J.-L. Fattebert and F. Gygi, Comput. Phys. Commun. **162**, 24 (2004).
[20] O. Tatebe and Y. Oyanagi, in *Supercomputing '94., Proceedings* (1994), pp. 194–203.
[21] S. Li, S. Ahmed, and E. Darve, J. Comput. Electronics **6**, 187 (2007).
[22] L. Lin, C. Yang, J. Lu, L. Ying, and E. Weinan, SIAM J. Sci. Comput. **33**, 1329 (2011).
[23] K. Brandhorst and M. Head-Gordon, J. Chem. Theory and Comput. **7**, 351 (2011).
[24] S. Demko, W. F. Moss, and P. W. Smith, Math. Comp (1984).
[25] M. Benzi, , and N. Razouk, Elec. Trans. Num. Anal. **28**, 16 (2007).
[26] K. Chen, *Matrix Preconditioning Techniques and Applications* (Cambridge University Press, Cambridge, UK,, 2005).
[27] L. Kolotilina and A. Yeremin, SIAM J. Matrix Anal. Appl. **14**, 45 (1993).

[28] M. Benzi and M. Tuma, SIAM J. Sci. Comput. **19**, 968 (1998).
[29] Y. Saad, SIAM J. Sci. Comput. **14**, 461 (1993).
[30] Y. Saad, *Iterative Methods for Sparse Linear Systems, 2nd edition* (SIAM, Philadelpha, PA, 2003).